Course Introduction

Dr. Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science
Welcome to CS 421!

Topics for discussion:
- Logistics – instructor, course objectives
- Why study languages?
- Major themes for the course
Me!

Name  Mattox Beckman

History  PhD, Fall 2003, University of Illinois at Urbana-Champaign
Lecturer 2003–2015 Illinois Institute of Technology

Research Areas  CS Education, Programming Languages, Mathematical Foundations of Computer Science

Specialty  Partial Evaluation, Functional Programming

Professional Interests  Teaching; Computer Science Education; Functional Programming; Semantics and Types; Category Theory

Personal Interests  Cooking; Go (Baduk, Wei-Qi, Igo); Philosophy; Evolution; Meditation; Kerbal Space Program; Home-brewing; ... and many many more ...
Specification Learning

- The class will not be graded on a point system. We will use Specifications Grading instead.
- There are 20 “learning modules”, most with 4–5 outcomes. Each outcome is measured by an assessment. Your grade is based on how many learning modules you complete.
- Assessments are graded pass/fail, and most will have significant retry opportunities. To complete a learning module, you must get 80% of the available points.
- Three sources:
  - Prairielearn Activities — given after class. Up to 50% of the points available this way. (More on these next slide.)
  - Exams — assessments show up on midterms, and again on the final. All exams will have a practice exam.
  - Machine Problems — some machine problems fulfill part of a learning module.
Activities

- This is a **flipped** classroom!
  - Please watch the lecture video *before* coming to class!
- In class activities POGIL to reinforce learning.
- Prairielern activities to consolidate/apply learning.
- There is not necessarily a post-class activity for each day.
POGIL

- Process Oriented Guided Inquiry Learning
- Based on > 20 years of research in how students best learn.
- Four roles:
  - Manager: watches time, keeps team on task, etc.
  - Recorder: will fill out the worksheet
  - Reporter: asks questions on behalf of group, communicates to class
  - Reflector: observes how team performs
- We will use breakout rooms; probably will just randomize each time.
Machine Problems

- Designed to help you apply what you’ve learned, and to achieve major course objectives.
- You are allowed one partner for the programming part, but **you must cite your sources!** (Place partner netids in a comment at the top.)
- Don’t use the “perturbation method” of solving machine problems! We expect you to _understand_ the solution and the process very well.
- See the syllabus for more details.
Exams/Quizzes

- The purpose of an exam is to measure mastery of material.
  - Exams are subdivided into proficiency units. These count toward your learning modules.
  - The final exam will repeat all the proficiency units.
- Three midterms, held at 19:00 (7pm), two hours long.
- One final
Why Study Languages?

- Pai sei
- Blub – see *Beating the Averages* by Paul Graham. [Gra03]
- Language families
Pai Sei

Different languages can express different concepts efficiently!

▶ A story from human languages: pai sei
Pai Sei

Different languages can express different concepts efficiently!

- A story from human languages: *pai sei*
- Languages and cultures grow together to shape each other.
Different languages can express different concepts efficiently!

- A story from human languages: *pai sei*
- Languages and cultures grow together to shape each other.
- It’s difficult to reason about something without vocabulary!
Different languages can express different concepts efficiently!

- A story from human languages: 
  - *pai sei*
- Languages and cultures grow together to shape each other.
- It’s difficult to reason about something without vocabulary!
- See *Politics and the English Language* by George Orwell. [Orw46]
Blubs

▶ From *Beating the Averages* by Paul Graham
Introduction and Logistics

Assignments

Course

Blubs

► From *Beating the Averages* by Paul Graham

► The difference between a known powerful language to a less powerful language is easy to see.
From *Beating the Averages* by Paul Graham

- The difference between a known powerful language to a less powerful language is easy to see.
- The difference between a known less powerful language to a more powerful language is not easy to see!
Themes

The course has four major parts:

1. Functional Programming
   You will learn functional programming by learning how to build interpreters in HASKELL.
Themes

The course has four major parts:

1. Functional Programming
   You will learn functional programming by learning how to build interpreters in HASKELL.

2. Parsing
   You will learn how text becomes a data structure we can use to represent a program.
The course has four major parts:

1. Functional Programming
   You will learn functional programming by learning how to build interpreters in HASKELL.

2. Parsing
   You will learn how text becomes a data structure we can use to represent a program.

3. Mathematical Foundations
   You will learn some of the mathematical theory that lets us reason about programming languages and the programs written in them.
Themes

The course has four major parts:

1. Functional Programming
   You will learn functional programming by learning how to build interpreters in HASKELL.

2. Parsing
   You will learn how text becomes a data structure we can use to represent a program.

3. Mathematical Foundations
   You will learn some of the mathematical theory that lets us reason about programming languages and the programs written in them.

4. Pragmatics
   You will learn some of the design decisions available to you when choosing (or creating!) a language.
So, what should you learn?

- Understand major classes of programming languages: techniques, features, styles.
- How to select an appropriate language for a given task.
- How to read a formal specification of a language and implement it.
- How to write a formal specification of a language.
- Some Powerful Ideas:
  1. Recursion
  2. Abstraction
  3. Transformation
  4. Unification

The emphasis is on learning the theory, knowing why the theory is valuable, and using it to implement a language.
Bibliography

